Misinformation detection in news text: Automatic methods and data limitations

Fatemeh Torabi Asr, Mehrdad Mokhtari, Maite Taboada[*]

Discourse Processing Lab, Simon Fraser University

Abstract

We address the difficulty of automatic misinformation detection through state-of-the-art natural language processing techniques. Machine learning and natural language processing are often touted as the perfect solutions for the problem of detecting misinformation at scale. We argue, however, that current approaches tend to fall short because of the unavailability of reliably annotated data. Given the scarcity of quality labelled data, we first conduct a data collection effort by leveraging fact-checking websites. Second, we perform a comparative feature analysis of the news articles with true vs. false content. Finally, we conduct a set of text classification experiments using a variety of methods and show that the quality of the training data in terms of its labelling system and balanced coverage of topics directly affects the classification accuracy on test data (news articles that the classifier did not see during training). Feature analysis experiments show specific trends of linguistic patterns in fake news articles. In predictive classification, some of these features such as n-grams and semantic features help considerably in recognizing false from true news articles and some features such as readability features tend to be less helpful. We also compare deep learning classification models against feature-based models and show that, given the small size of currently available data, feature-based models are more capable of cross-topic generalization. This result points to the need for automatic classification methods that are informed by linguistic, corpus linguistic, and stylistic research. Finally, we show that using data labelled based on the reputation of the sources does not result in accurate classification of test data, which, in turn, motivates future data collection with reliable tagging and on diverse topics.

Keywords: misinformation, fake news, text classification, natural language processing (NLP), machine learning, labelled datasets, corpus linguistics

1. Introduction: The problem of detecting the language of misinformation

A major vulnerability of public discourse online—a fast, unbound and often anonymous medium—is the increased susceptibility to deception. When we started this research, in 2017, 'fake news' was an accusation against some news organizations and a small area of research in the academic community. Just a few short years of political turmoil across the globe, a pandemic, and a climate crisis have brought misinformation to the forefront of public discourse and to the attention of many researchers.

Fake news has been defined in different ways, such as "news articles that are intentionally and verifiably false, and could mislead readers" (Allcott and Gentzkow, 2017). We find 'misinformation'

[*] To appear in: Maci, S., M. Demata, M. McGlashan and P. Seargeant (eds.) The Routledge Handbook of Discourse and Disinformation. Routledge. This version: May 2022. Corresponding author: mtaboada@sfu.ca

(wrong information) and ‘disinformation’ (wrong information with the intention to deceive) more accurate terms for scientific research as compared to ‘fake news’, which is frequently used in political discourse and media stories (for definitions, see Habgood-Coote, 2019; Tandoc Jr et al., 2018; Wardle and Derakhshan, 2017). The subject of our study is false information in news text, regardless of the distributing source’s intention, thus misinformation in its general sense, which can manifest itself as rumours and hoaxes, propaganda, or even false information in mainstream news publications.

The research community turned its attention to the phenomenon in 2015 and 2016 (Connolly et al., 2016; Perrott, 2016), with two comprehensive studies published in 2018 (Lazer et al., 2018; Vosoughi et al., 2018). The latter in particular clearly established the danger of misinformation: Fake news stories are particularly dangerous because they not only tend to reach a larger audience, but also penetrate into social networks nearly 10 times faster than fact-based news (Vosoughi et al., 2018).

The current trends to combat the misinformation problem take three main approaches: educate the public, carry out manual checking, or perform automatic classification. Educating the public involves encouraging readers to check the source of the story, its distribution (who has shared it, how many times), or to run it by fact-checking websites. This is certainly necessary, but it will not be enough. Organized manual checking, before or after publication, is a possibility, but it is also not a realistic solution, given the fast spread of misinformation that Vosoughi et al. (2018) found. Approaches from machine learning, computational linguistics, and natural language processing (NLP) show promise, in that they can perform automatic classification and can help complement the efforts of fact-checking sites. The promise is that we will be able to detect fake news stories automatically, before they have a chance to spread and do harm. The process of fact-checking can be modeled as a series of NLP tasks, from identifying claims and rumours to comparing information and producing fact-checking verdicts and justifications (Guo et al., 2022). In this paper, we explore the deployment of a specific NLP task, text classification.

One of the important challenges in automatic misinformation detection using modern NLP techniques is data (Asr and Taboada, 2018, 2019). Annotation of fake news is a resource-demanding and particularly sensitive task because of the wide spectrum of public opinions about who exactly is a reliable source, including established news organizations. The majority of automatic systems built to identify fake news rely on training data (news articles) labelled with respect to the credibility or the general reputations of the sources, i.e., domains/user accounts (Fogg et al., 2001; Horne et al., 2018; Nørregaard et al., 2019; Rashkin et al., 2017; Volkova et al., 2017; Yang et al., 2017). Even though some of these studies try to identify fake news based on linguistic cues, what they eventually model is the publisher’s general writing style (e.g., common writing features of the publishing websites) rather than the linguistic similarities of the articles containing false information.

For example, Rashkin et al. (2017) collected news articles from websites that they categorized as general publishers of Hoax, Propaganda, Satire or Trusted (mainstream) news. They showed that a classifier trained on news articles from some of these websites could identify news from other websites from the same category, thus learning the general linguistic characteristics of each type of publisher. Detecting the style of a news article in terms of belonging to coarse categories such as Satire, Propaganda, Hoax or Trusted mainstream outlets is an interesting task, but not exactly what we would like to do in our battle against fake news. The goal of our paper is to pursue a slightly different and hypothetically more difficult task, namely detecting, based on linguistic properties, whether or not a news article contains false information. This is useful, because the approach could then work across different sites, regardless of publisher.

In terms of methodology, we focus on a content-based approach to news text classification. Rather than using contextual metadata such as user activity features, network cues, or credibility of the publishing sources, we assess the feasibility of detecting misinformation by examining the content of the article, i.e., the text itself. This puts our work in the category of *style-based* fake news detection, as opposed to *context-based* or *knowledge-based* detection (Potthast et al., 2018) and in the area of language-based detection (Lugea, 2021). The hypothesis behind our approach is that deception in news has its own style, i.e., a language for misinformation. If the language of news articles with true vs. false content is different, then we should be able to detect misinformation even with no access to metadata or a universal knowledge base about which facts are true.

In order to investigate this hypothesis, we explore a variety of data collections and state-of-the-art text classification techniques. We test both classic feature-based models and modern deep learning classifiers. Big data is required for robust performance and especially in the case of deep learning models. Unfortunately, however, available datasets for automatic misinformation detection are either small in size but accurate in labels, or large in size but labelled based on source reputation. To address the lack of data, we leverage fact-checking websites and collect news articles that these fact-checkers have labelled as false and true. These collections are still much smaller than standard benchmark datasets in text classification. Therefore, we also test two transfer learning approaches:

- Label transfer: using a large dataset of reputation-based labelled news articles as training data and considering news articles from generally-known fake publishers as “false” and that of mainstream publishers as “true” (mapping Propaganda, Hoax and Satire in Rashkin’s data to “false” and the Trusted category to “true”).

- Knowledge transfer: using a relatively small dataset of fact-checked news articles as training data in combination with a pre-trained language model based on deep learning techniques for text classification.

In both training settings, we tune model parameters on a validation set and then test on reliably labelled news articles, which have been individually fact-checked and rated as true or false. Our test datasets are either sampled from the same distribution of the training data (mixed claims/topics/headlines) or a different distribution (unseen claims/topics/headlines). Our experiments show that a ‘classic’ feature-based model trained on the small but fact-checked dataset would generalize better and achieve a higher accuracy on unseen topics. The knowledge transfer model, which uses a pre-trained language model as its linguistics backbone, can fine-tune itself to the small training data and achieve a higher accuracy on test data from a similar distribution. However, this model does not generalize well to unseen topics. Finally, the label transfer technique that uses large training data collected from categorized publishers does not distinguish between false and true news articles in a fact-checked test dataset and has a large bias towards labelling anything as fake news. This suggests that reputation-based labelling, despite its potential to easily provide us with big data, is not suitable for misinformation detection in terms of veracity checking. Error analysis reveals that, for a robust and scalable classification, we would need more fact-checked news articles from a variety of topics and balanced across labels.

The contributions of this approach are twofold, in data and in methods. In data, we explore the right amount and type of data necessary to reliably train misinformation classification methods, and release a mid-size dataset for the task. In terms of methods, we first conduct a feature analysis on false and true news articles and then build and evaluate a variety of text classifiers for predictive modeling. Above all, we see these experiments as a cautionary tale on the use of NLP and machine learning techniques on data that has not been properly collected and examined.

2. Related work

Many of the studies in detecting misinformation apply what we may call ‘classic’ machine learning methods, i.e., supervised classification with a variety of features. The features range from surface characteristics such as document length and n-gram frequency to specific types of semantic classes (e.g., subjectivity and emotion markers), syntactic features (e.g., depth of syntactic tree and frequency of each part of speech) and discourse-level features (Afroz et al., 2012; Conroy et al., 2015; Horne and Adali, 2017; Pérez-Rosas and Mihalcea, 2015; Rashkin et al., 2017; Rubin et al., 2015; Ruchansky et al., 2017; Volkova et al., 2017). Some of these studies have been characterized as stylometric (Przybyla, 2020), in that they use the style of the language as an indicator of misinformation.

Algorithms deployed in this type of supervised learning are often Support Vector Machines (SVMs), with a feature engineering and feature selection process. Performance in these approaches tends to plateau as data increases, showing that features are useful with smaller amounts of data, but performance increases stall at some point as amount of available data increases. Therefore, these methods are considered to have an important limitation (Ng, 2011).

A second set of studies use modern neural network models. In cases where large amounts of data are available, deep neural network models tend to achieve more impressive results. Deep learning has, in general, taken over many natural language processing tasks, at least in domains where large-scale training data is available. Deep learning models in NLP usually rely on word vectors and embedded representations. Although it is possible to extract embeddings from domain data, most methods rely on pre-trained embeddings (Le and Mikolov, 2014; Pennington et al., 2014). Models in deep learning include Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs) and Attention models (Conneau et al., 2017; Lai et al., 2015; Le and Mikolov, 2014; Medvedeva et al., 2017; Yang et al., 2016; Zhang et al., 2015). They perform slightly differently, depending on the task and the type of data. In general, the task tends to be a binary classification task (i.e., is this text X or Y?), in our case whether the text in question is an instance of fake news/misinformation or an instance of reliable, fact-based news. For this task, what RNNs do is encode sequential information in the articles, modeling short text semantics. CNNs are composed of convolution and pooling layers, providing an abstraction of the input. CNNs are useful in tasks where presence or absence of features is a more distinguishing factor than their location or order, and work well when classifying longer text. For instance, CNNs are helpful in sentiment analysis of product reviews, where the distinguishing features between positive and negative reviews may be the relative frequency or presence of positive and negative words (Dos Santos and Gatti, 2014; Kucharski, 2016; Ouyang et al., 2015).

The issue with many of these studies is the data collection methodology. Many of the existing datasets use publisher/website reputation as the main criterion for collection (Przybyla, 2020). A website is assumed to be a "fake news publisher" as a whole, with no individual labelling of data. It is often the case, however, that these publishers mix truth with lies, publishing relatively true stories, or republishing stories from reputable sources. It is also the case, albeit much less frequently, that reputable websites may inadvertently publish inaccurate information (Fichtner, 2018; Mantzarlis, 2017). In fact, this fundamental distinction between focusing on fake news outlets vs. fake news stories may the source of much confusion about how much misinformation is actually circulating (Ruths, 2019).

The solution to the unreliability of publishers is to label each news article individually as to whether it is reliable or not. This kind of data is rare, and it is difficult to assign such labels, as the task requires professional expertise and background knowledge.

Small existing datasets include a collection by Allcott and Gentzkow (2017), who annotated 156 articles by manually consulting three fact-checking websites (Snopes, Politifact and Buzzfeed) and downloading the source page of the debunked rumors and the 40 articles annotated by the Credibility Coalition (Zhang et al., 2018). Most other data is either short statements rather than full articles (Wang, 2017), satirical news articles rather than misinformation (Rubin et al., 2016), annotations for stance (Thorne et al., 2018), or articles that were modified to be made untrue (Pérez-Rosas et al., 2017). The dataset from Shu et al. (2020) comes closest to the requirements for this task, as it was crawled from fact-checking websites (but not validated after scraping).

In summary, although a few different types of datasets exist, none of them contain a large enough number of both fake and legitimate news articles, which is the type of data that we need to learn to classify misinformation (as opposed to classifying stance, headlines or satire). For this reason, we collected our own data, using fact-checking websites, as we describe in the next section.

3. Quality data for misinformation detection

We have established that a reliably labelled collection of false and true news articles needs to be individually labelled, not scraped from specific websites. To achieve this individual labelling, we leverage fact-checking websites to collect news articles with false and true content. We use:

- a dataset of news articles on the topic of US election in 2016 from a BuzzFeed study; and
- a dataset that we collect by harvesting the Snopes fact-checking pages and downloading the source text of the news articles.

3.1. BuzzFeed data

The first source of information that we used to harvest full news articles is a BuzzFeed collection of links to Facebook posts, originally compiled for a study around the 2016 US election (Silverman et al., 2016). It includes links to posts from nine Facebook pages (three right-wing, three left-wing and three mainstream publishers) and manual annotation of the veracity of individual posts by an instructed group of raters. The dataset of links with user interaction information is available via Kaggle (`https://www.kaggle.com/mrisdal/fact-checking-facebook-politics-pages`). Unfortunately, only the Facebook link is available there, not the full article. We took each individual link, followed it to the Facebook page, and then followed the link to the original post. We scraped the webpage of the original post to extract the full text of each article, which we cleaned of HTML tags and any other extraneous material. Veracity labels come in a 4-way classification scheme including *mostly true*, *mixture of true and false*, *mostly false*, and *containing no factual content*. This dataset contains 1,380 articles. We refer to this collection as the *Buzzfeed USE* dataset.

We also use a collection of Buzzfeed selected top fake news of the year 2017 for test purposes in our classification experiments. A similar scraping process was performed to get the content of the news articles in this collection. We refer to it as the *Buzzfeed Top* dataset, and it contains 33 news articles with false content (note that this dataset is not balanced, as it contains only fake news stories). Both of the Buzzfeed datasets have been published in our previous work (Asr and Taboada, 2018, 2019). The following dataset is a new contribution.

3.2. Snopes data

The second source we used for collecting news texts with veracity labels is the Snopes fact-checking website. Snopes is one of the oldest and most well-known rumor debunking websites, run by a team of expert editors. In addition to finding rumors and mentioning distributing sources, they provide elaborate explanations of the rumor and its effects. News articles collected from Snopes come with a fine-grained labelling (*true*, *mostly true*, *mixture of true and false*, *mostly false*, and *false*). The diagram in Figure 1 shows the process of data collection from Snopes fact-checking pages.

The challenge with using Snopes is that the site does not reproduce the entire text of the false or debunked news article; it instead publishes an article discussing a false story or rumor. The structure

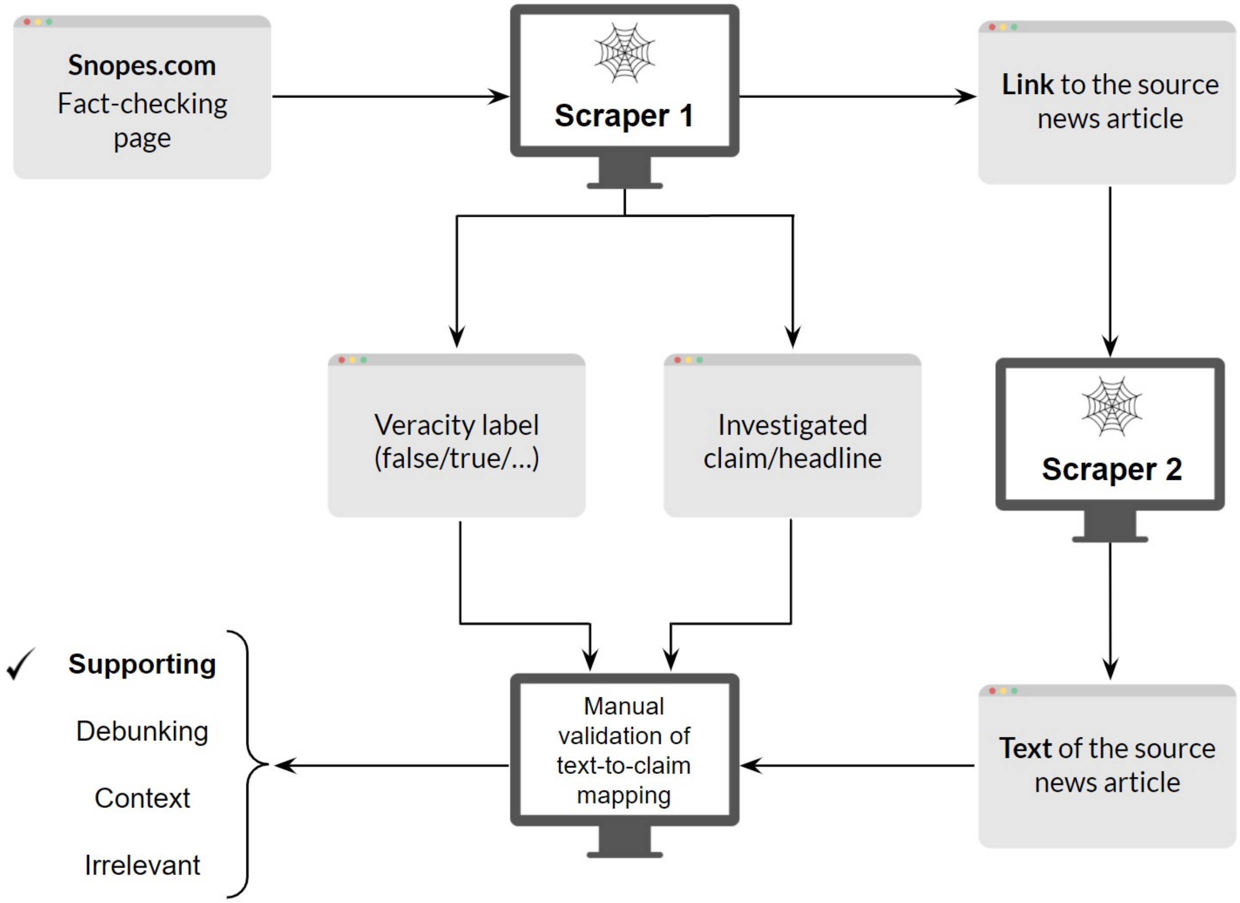
Snopes.com
Fact-checking page
Scraper 1
Link to the source news article
Veracity label (false/true/...)
Investigated claim/headline
Scraper 2
✓ Supporting
Debunking
Context
Irrelevant
Manual validation of text-to-claim mapping
Text of the source news article

Figure 1: Data scraping and validation process

of a Snopes article is the investigation of a claim, rumour, or story that has been contested. Snopes then discusses the content of that contested story, including a number of links, some to the source of the false story, some to sites that debunk the story or other links for extra information on the topic. Our task, then, was to follow those links and, for instance, in the case of a "false" label on the Snopes site, to find the link containing the full article with the false story. The entire process involved, for each claim ("true" or "false"), scraping the discussed claim, the veracity scoring of the claim according to the Snopes labelling system, and the links to the source of the claim. Given that the scraped data might be noisy (e.g., other links on the webpage might be harvested but do not necessarily point to the source news article), a verification step is required afterwards.

We first cleaned the data by removing exact duplicates, texts that were too short, noisy (i.e., advertisements) or texts that were scraped from outdated links (pages showing a message like "this content has been removed"). We then verified the alignment between the discussed claim and the scraped body text of each article by crowd sourcing annotators on the Figure Eight platform (now part of Appen). During this process, we gave more than 4,000 texts and supposedly associated headlines to the annotators and asked them to select the alignment of the text with respect to the headline. If the alignment was right, they would next label the body text (which we scraped from the source article) as *supporting* the given headline (which we scraped from the Snopes fact-checking page). Otherwise, they would choose one of the other answers (*debunking* the headline, *context* information, or *irrelevant* text). Finally, we filtered all items that had the "supporting" label, which were the actual source of the debunked rumors. We refer to this dataset as the *Snopes Silver* dataset, because it is annotated by crowd workers rather than in-house experts (which would make it Gold; see below). The data contains 2,075 articles (1,425 false, 160 mostly false, 231 mixture, 17 mostly true and 242 true items). This dataset is publicly available on our lab repository (`https://github.com/sfu-discourse-lab/MisInfoText`).

Before conducting the crowd sourced experiments on the automatically collected Snopes articles, we picked a semi-randomly selected set of the scraped articles and verified their association with the claims and veracity labels with the help of two annotators in our lab. Two annotators performed independent assessment of the claim and the article supposedly containing that claim. A third annotator, one of the authors of this paper, examined each claim and article and solved any disagreements. This data includes a total of 118 news articles (17 false, 27 mostly false, 26 mixture, 26 mostly true and 22 true items). We refer to this dataset as the *Snopes Gold* dataset, as it was more carefully curated within our lab. We have also referred to a superset collection containing these items as the Snopes312 dataset in previous work (Asr and Taboada, 2018, 2019). We made sure that the Gold dataset included a balanced number of articles from each veracity label and that the claims/news headlines in this smaller set did not overlap with our Silver dataset. These decisions were made so we could use the Gold dataset both as training material for crowd-source annotation and as test data (unseen claims) in our automatic text classification experiments, which will be presented in the following sections.

When we use these datasets in our experiments, we remove the “mixture” items and build a false and true category by collapsing the finer grained distinctions. This gives us a total of 1,649 false and 1,349 true items.

4. Analyzing linguistic features in false and true news

The first step towards understanding the linguistic differences between false and true news articles involves conducting a feature analysis. In this section we look closely at the linguistic difference between the two datasets of news text that we introduced in the previous section and also the differences between the true and false items in each dataset. In order to conduct a feature analysis experiment, we studied five main categories of linguistic features ranging from surface properties of the text to measures of readability.

- **Surface text features.** We consider the surface properties of a text such as length and punctuation frequency. Previous work in text classification has found correlations between such features and whether the text is an instance of fake or mainstream news (Biyani et al., 2016; Horne and Adali, 2017). We employ the following surface properties of the text: number of characters, words and sentences; proportion of punctuation; uppercase characters; and average sentence length in terms of number of words.

- **N-grams.** Short sequences of words, known as n-grams, have a long history in NLP. In text classification, models using n-gram features have been vastly successful and they have set a difficult baseline even for modern deep learning models (Zhang et al., 2015). They represent an approximation of phrases appearing in a text and their frequencies across documents within a corpus can be informative with regard to the similarities and differences across texts. We use the TF-IDF (term frequency-inverse document frequency) scored n-grams generated by the *scikit-learn* python library to visualize and understand the topic differences between fake and real news articles.

- **Semantic features.** The frequency of words coming from a specific semantic category, such as negative polarity words, words related to religion, sex, feelings, body, money and work can provide useful information about a text. Studies on deceptive text and fake news have used such features to try and find distributional differences between lies and truthful statements (Biyani et al., 2016; Pérez-Rosas et al., 2017; Pérez-Rosas and Mihalcea, 2015; Rashkin et al., 2017; Rubin et al., 2016). We consider a large set of semantic lexicons, where words are assigned to specific semantic categories, in order to extract semantic features from each text document. We include all lexicons from the *LIWC* (Linguistic Inquiry and Word Count) inventory (Pennebaker et al., 2015) as well as a set of lexicons for markers of subjectivity and biased language (Recasens et al., 2013; Wilson et al., 2005)

- **Syntactic features.** Syntactic properties of text have been used for text classification in a variety of domains including deception detection (Mukherjee et al., 2013; Pérez-Rosas et al., 2017; Pérez-Rosas and Mihalcea, 2015; Post and Bergsma, 2013). We use the proportional frequency of each part of speech tag from the universal POS tagset of the NLTK library. This tagset includes 12 general tags including NOUN, PRON, ADJ, ADV, VERB, ADP, NUM, PRT, DET, X, CONJ, and punctuation.

- **Readability features.** Measures of text coherence such as readability indices have shown to be helpful features for text classification and particularly in deception detection (Pérez-Rosas et al., 2017; Pérez-Rosas and Mihalcea, 2015). We extracted the readability indices of each text with the help of the *textstat* python library, which gives us a number for each of the following measures: Flesch reading ease, Smog index, Coleman-Liau index, Linsear write formula, Dale-Chall readability score, automated readability index, Flesch Kincaid grade, and Gunning Fog score.

In order to provide a clear overview of the important features, we compute each of the above feature sets for the Buzzfeed and Snopes datasets separately. We expect overlapping patterns between the two datasets in terms of the differences between the false and true news instances.

For each dataset, We first mapped all “false” and “mostly false” items to one label, i.e., “false”, and “true” and “mostly true” to “true”. We also removed “mixture” items to avoid basing our analysis on edge cases. We then combined all documents of the false class and all documents of the true class so we can compute the average value of each feature for each class of news articles. We then provide two quantitative measures to discuss the importance of a given feature: 1) the proportional average value of the feature in false to true news articles, and 2) the p-value of a correlation analysis that reveals whether the feature value difference between the false and true categories is statistically significant. We compute the p-value based on the Recursive Feature Elimination (RFE) method of the scikit-learn python library, applied to each feature set separately, and present the most discriminating features ($p - value < 0.001$) in the result tables.

Table 1 shows some of the features that help distinguish between false and true news articles. All listed features in this table are significantly more frequent in one of the two categories. The rows within each feature set are sorted in ascending order based on the ratio of occurrences in false news articles to that of the true news articles within the Snopes dataset. In this table, we only kept rows that had the same pattern in the Buzzfeed dataset to avoid confusion. Two separate tables for each of the two datasets are available in the Appendix. Shaded rows in the table refer to the general features of false news articles and rows with no background colour are those that occur more often in true news articles.

In general, the textual features reveal that false news in both datasets were on average shorter than true news and contained fewer punctuation marks. However, in the Buzzfeed data only the proportion of punctuation came out as a statistically significant marker. Therefore, other surface features are not listed in the above table.

Table 1: Important features for distinguishing true from false news articles; only overlaps between Buzzfeed and Snopes Silver datasets are included here. Shaded rows are features of false news articles.

Feature category	**Feature**	**Ratio false/true**	**P-value**
Surface	num_punc/num_char	0.926844562	0.001331224
Semantic lexicon	comparative_forms	0.947737051	0.020227509
	negative_HuLui	1.133089752	0.00267177
	negative_mpqa	1.192098625	0.000864263
	modal_adverbs	1.298092007	0.005319897
	manner_adverbs	1.35698137	0.001742446
Semantic LIWC	apostro	0.648061274	0.002444304
	work	0.876750931	0.00999914
	Sixltr	0.940804067	0.033111439
	cogproc	1.06101936	0.057419697
	pronoun	1.070951735	0.029411617
	auxverb	1.077592026	0.093201528
	adverb	1.110034072	0.00311221
	they	1.151274939	0.066116507
	bio	1.224761165	0.041183623
	anx	1.383614276	0.097296795
	sexual	2.753109267	0.060287064
Syntactic	NOUN	0.970826348	0.021800144
	VERB	1.034897463	0.008362629
	PRT	1.040020625	0.097773857
	ADV	1.093310376	0.000661848
	PRON	1.094581886	0.011254044

With the large number of semantic features considered in our analysis, it is not surprising to see a handful of them coming out as distinguishing features for content veracity. We find both subjectivity markers and LIWC properties in the list of helpful features. Among the lexicon-based features, we find more negative polarity words in false news compared to true news. We also find a relatively larger proportion of adverbs, and in particular, modal and manner adverbs in false news. A higher usage of comparative forms is, however, a marker of true news. Based on our analysis of LIWC features, words related to sex, death, anxiety, biological, and cognitive processes frequently occur in false news articles; whereas in true news, we found a larger proportion of words related to work and money. These patterns reveal how true and false news articles may differ in the distribution of topics and headlines. While true news articles usually discuss serious and more abstract topics, false news is more focused on topics that can quickly capture the reader's attention. Texts in the true category seem to contain longer words (Sixltr means word with more than 6 letters) and more apostrophes (previously also captured as more punctuation in true news). These features may reveal the more sophisticated writing style of true news, which normally appear on a moderated website with editor supervision.

Finally, the syntactic feature analysis shows that true news contains a larger proportion of nouns and numbers (the latter only significant in Snopes data); whereas false news contain more adverbs, particles and pronouns. Looking more closely at the usage of pronouns, we find that in false news most pronouns refer to the plural third person "they", whereas in true news the first person pronoun "I" is relatively more frequent (the latter pattern significant only in Snopes data).

This can be attributed to the use of direct quotations or self-mention of the reporter in true news, as opposed to repeated mention of other entities in false news, perhaps a sign of othering certain groups of people (Riggins, 1997).

Extracting the most important n-grams is a challenging task due to the large number of such features. Moreover, by examining the individual n-grams one can hardly infer a general pattern regarding the differences they reveal between false and true news articles. In order to examine the n-grams most specific to each dataset, we first combined the Snopes Silver and the Buzzfeed USE corpora and then extracted 100 n-grams that occurred in less than half of all the documents (to avoid corpus-specific stop words) but in more than three documents within the combined corpus. We then analyzed these unigrams the same way as other feature types: we calculated the false to true proportion of each unigram and filtered those with highest and lowest values and a $p-value < 1$ according to the Recursive Feature Elimination method. The list of most discriminating unigrams based on this technique is provided in Table 2.

The majority of high-score unigrams marking the true news articles are focused around the topic of the US election. This is expected, given the fact that most true examples in the combined dataset come from the Buzzfeed USE corpus, which includes news related to the US presidential candidates and events around the 2016 election. The Snopes data includes a variety of topics and most false articles come from this dataset; that is why the high-score unigrams in the false class come from a more diverse and general vocabulary, as it is evident from the table. Now, this imbalance in terms of topic vocabulary between the two classes of news articles may raise a challenge for building predictive models based on the presented data: If we train a model on this dataset, the classifier may overfit to fine-grained lexical features rather than high-level properties of the text and this may result in weak generalization and low accuracy on collections of false/true news articles with a different topic distribution. We will discuss this further in the next section.

5. Misinformation detection through text classification

A variety of different machine learning techniques have been applied to text classification problems such as sentiment analysis, product review classification, authorship recognition, and deception detection in text (Eisenstein, 2019; Hovy, 2020). These methods include classic feature-based classification algorithms and deep neural network models, which instead of using features exploit pre-trained word embeddings or language models. Deep learning models usually achieve a higher accuracy compared to traditional classifiers such as Support Vector Machines using engineered features, but they require more training data to converge. In a situation where training data is scarce, such as the case of misinformation detection, classic feature-based algorithms may be preferred over deep learning. For example, Zhang et al. (2015) showed that the classic TF-IDF model worked better than a variety of deep learning models if less than a million training records were available.

In this section, we explore the performance of different text classification techniques when used in combination with our data for detecting false from true news articles. We employ both a classic method, i.e., a Support Vector Machine classifier with linguistic features and a deep learning model, BERT (Devlin et al., 2019). We train them on different slices of our collected data and examine their generalization power on several test datasets. We also consider a setup with larger training data that has been labelled based on the reputation of the publishing sources rather fact-checking of each individual news article. These experiments will help us reveal the pros and cons of each classification technique and draw future directions especially for collecting more and better quality data.

Table 2: TF-IDF unigram features with highest proportion in true (top) vs. false news (bottom) within the combined corpus (Snopes Silver and Buzzfeed USE)

Feature	Avg. in true	Avg. in false	Ratio false/true	P-value
debate	0.066639	0.011568	0.173589	1.18E-50
voters	0.043444	0.009592	0.220792	2.48E-31
clinton	0.114134	0.031633	0.277158	2.71E-72
presidential	0.065025	0.019048	0.29294	8.90E-56
campaign	0.075448	0.022253	0.294945	1.77E-52
republican	0.062131	0.019052	0.306639	1.74E-46
hillary	0.074586	0.023268	0.311959	5.28E-53
donald	0.087009	0.032008	0.367873	1.74E-62
trump	0.150655	0.063265	0.419935	4.73E-67
vote	0.036512	0.018449	0.505276	1.03E-08
election	0.041777	0.021478	0.514117	7.42E-12
today	0.025932	0.036286	1.399304	4.85E-04
year	0.044547	0.06537	1.467454	8.41E-10
come	0.02743	0.040592	1.479861	3.47E-06
family	0.029711	0.044435	1.495613	2.58E-05
home	0.026483	0.041189	1.555327	9.89E-06
school	0.021147	0.034088	1.611911	2.88E-04
world	0.03383	0.056416	1.667639	5.27E-10
use	0.027951	0.049615	1.775065	1.07E-09
children	0.023254	0.043545	1.872602	3.24E-08
old	0.024174	0.047418	1.96152	3.68E-13

5.1. Models

Feature-based model. We use a Support Vector Machine Classifier (SVM) from the scikit-learn python library with all the linguistic features that we introduced and analyzed in the previous section. These features include surface text features, TF-IDF scored n-grams, semantic category features, syntactic features (parts of speech counts), and readability scores. Both the TF-IDF vectorizer and the SVM classifier had a set of parameters that we tuned through cross validation on training data. The best values for parameters of the TF-IDF vectorizer were max-df=0.5, min-df=5, n-gram-range=(1,2) and sublinear-tf=True. Best parameter values for the SVM were penalty="l2", tol=1e-3 and others set to default.

Deep learning model (BERT, Bidirectional Encoder Representations from Transformers). In order to apply deep learning to the task of misinformation detection, we use a well known language representation model in sentence classification tasks such as sentiment analysis or fact-checking. BERT, which stands for Bidirectional Encoder Representations from Transformers (Devlin et al., 2019), is based on a Transformers architecture (Vaswani et al., 2017). BERT-base, the model that is used in this article, comprises 12 transformer layers, where every transformer layer accepts a list of token embeddings, and generates the same number of embeddings with the same hidden size on the output. The output of the last transformer layer of the classification token (a special token that is added for classification purposes) will be applied to feed a classifier. We use the PyTorch implementation of the pre-trained BERT model in Python using the transformer library of *Hugging Face* (Wolf et al., 2019). The BERT model was trained on Wikipedia articles with 1 million steps for 40 epochs, with batch size of 256 on 8 GPUs for 6.5 days. Due to the large number of parameters and layers in a BERT network and the high risk of overfitting, we fine-tune this pre-trained BERT model. To fine-tune, we train the entire pre-trained model on our training data and feed the output to a softmax classifier to compute logits. We optimize the main hyperparameter values, including the dropout rate, batch size, optimizer learning rate, and the number of epochs based on the average Area Under Curve (AUC) score from cross-validation on training data. The best values for the parameters of the BERT-based model were dropout=0.35, batch_size=12, learning_rate=1e-5 (Adam optimizer). We utilized the maximum sequence length of 512 and tried between 4 and 15 training epochs. With the smaller training data, higher epoch numbers (around 12) gave us better results (lower validation and training loss). However, with larger training data size the performance plateaued after about 5 epochs, so we keep that as the accepted parameter. It is also worth mentioning that smaller training data are more prone to suffer from overfitting, resulting in larger validation losses with the same hyperparameter values. Table C.7 includes our best range of hyperparameters for the BERT-base model.

5.2. Data preparation

While building a predictive model for any detection task, it is important to spend time preparing data by removing noise and balancing the samples for the prediction classes. Our observation during feature analysis of the datasets showed a topic imbalance across datasets and, most importantly, across target classes (false and true news articles), as we have shown in other work (Asr and Taboada, 2019). Therefore, we decided to consider two different data scenarios in our text classification experiments. Table 3 shows how we sample the Buzzfeed USE and Snopes Silver datasets for preparation of training data with two approaches. We consider two training data scenarios, one with a small balanced training data and another with a relatively large but mixed training data. For sampling the small and balanced dataset, we randomly picked 64 items from each class within the Buzzfeed USE dataset, because its false class only contains this many items. This was in principle to make sure that data from a focused topic (the US election) is represented in both false and true classes in the balanced dataset. We take a similar approach in sampling from the Snopes Silver data, by picking 259 items from each class. The total number of items in our small and balanced dataset is 646 news articles. Second, we consider a larger training dataset, that is, we put together all true and false news articles from the Snopes and Buzzfeed datasets and then sample 1,300 items per class from this collection, which totals 2,600 news articles. This dataset is about four times larger than our small sample, but it is unbalanced with respect to the distribution of topics across false and true news articles.

As test data, instead of sampling from the same data sources, which may result in an artificially high accuracy, we consider three separate test datasets. These are all datasets that have been manually checked for the content veracity of individual news articles. The first obvious choice is the Snopes Gold data that we described in the section on data collection (Section 3). Apart from having been verified manually, this dataset has the nice feature of including non-overlapping news headlines with the Snopes Silver data. Performance on this dataset would tell us about the generalization power of a model to new topics and headlines.

The second test data is Buzzfeed Top, which contains the 33 top fake news of the year collected by the Buzzfeed agency from Snopes fact-checking pages (see Asr and Taboada (2018) for more details on this data). This dataset only includes false news and may have some overlap with topics in our training data, therefore should be an easy benchmark for all the models.

The third dataset is the collection of false and true celebrity related news articles collected and published by Pérez-Rosas et al. (2017), the Perez Celebrity dataset. This dataset has two nice characteristics: It is focused on the specific domain of celebrity stories, which may be less frequent

Table 3: Number of samples taken from each collection (Snopes Silver and BuzzFeed USE) to prepare our two different training datasets: the *small balanced* & *large mixed* samples

Dataset	False items	True items	Small balanced sample	Large mixed sample
Snopes Silver	$1,585$	259	$2 * 259$	$2 * 1,300$
Buzzfeed USE	64	$1,090$	$2 * 64$	
Total	$1,649$	$1,349$	646	$2,600$

in our training data, so it may reflect the generalization power of the models better; furthermore, this dataset contains a balanced number of false and true news articles on pre-selected matched topics (for instance, the same number of false and true articles about Jenifer Aniston's personal life!). Performance of our models on this last test dataset would be representative of cross-domain classification performance on real data.

As a separate kind of benchmark, we compare the performance of the same models on larger data that has been labelled based on the reputation of the publishing sources rather than based on the veracity of individual news articles. This method serves as the baseline in our experiments. We adopt Rashkin et al. (2017)'s dataset of Propaganda, Hoax, Satire and Trusted news articles, scraped automatically from websites based on their reputation. In order to map this data to our scheme of false and true news, we combine all items from Propaganda, Hoax and Satire and sample 4,000 news articles from this combination and label them as *false* news. We then sample 4,000 articles from the Trusted items and label them as *true*. That is why we call this data scenario the *mapped* data condition.

5.3. Experimental setup

In all the experiments across the three data scenarios, we first train a model on the training data and then test it on the three held-out test datasets. We report the performance of the model both on training data itself (to show the quality of the fit) and each of the test datasets by measuring the weighted F1-score (weighted average of the precision and recall for the false and true classes). As we mentioned before, parameter tuning is performed by cross-validation on training data prior to re-training on the entire set and the actual experiments.

For experiments using the SVM classifier, we present results on several feature sets to show the helpfulness of each feature category. We found a general trend that the TF-IDF n-grams were the best single set of features, followed by the semantic category features and the syntactic features. The readability and surface textual features came out as the least helpful ones for predictive modeling. In order to present the most relevant results, we picked the best feature setups and will report the outcome of these across different data scenarios.

The BERT model (Devlin et al., 2019), as introduced earlier in this section, is a state-of-the-art language model for NLP. This model was published in 2018 by researchers at Google AI. It contains 110M parameters and 12 transformer layers, which makes BERT training a hard task. Training a BERT model from scratch on a small set of data would greatly increase the likelihood of overfitting, the expensive computational costs aside. To avoid this problem, we take advantage of pre-trained BERT models that were trained on Wikipedia articles (`https://github.com/google-research/bert`). This language model is pre-trained on two NLP tasks: Masked Language Modeling and Next Sentence Prediction (Devlin et al., 2019). In this study, we fine-tuned the BERT pre-trained model in two different scenarios, however report results only for the second one. In the first approach, we froze the entire BERT architecture and attached one hidden-layer feed forward neural network as our classifier and trained this new model. During fine-tuning, we only updated the weights of the last attached layer. In the second method, we train the entire pre-trained model on our train data and feed the output to a softmax classifier to compute logits. The optimized hyperparameter values, which led to the results in Table 4, can be found in Appendix C.

Finally, we would like to mention that based on repeated experiments with a variety of parameter values with both the SVM and the BERT models, the final results depend also on the data sampling and the random effects present in both models. We report the most stable results in the paper rather than performing an aggressive brute force parameter tuning to obtain the highest possible score from a model.

Table 4: Performance of different text classification models in various data scenarios measured by F1-score on training data as well as three test datasets

Data scenario	Model	Train	BuzzTop	SnopesGold	PerezCeleb
Rashkin mapped (8,000 train items)	SVM N-gram	98	94	55	57
	SVM N-gram+Sem	96	94	52	57
	SVM N-gram+Sem+Syn	96	90	54	58
	SVM All features	92	86	56	57
	BERT	93	**97**	58	57
Large mixed (2,600 train items)	SVM N-gram	86	**97**	54	50
	SVM N-gram+Sem	85	**97**	54	50
	SVM N-gram+Sem+Syn	85	**97**	54	52
	SVM All features	83	85	50	52
	BERT	80	91	56	49
Small balanced (646 train items)	SVM N-gram	97	88	66	64
	SVM N-gram+Sem	93	88	**70**	**65**
	SVM N-gram+Sem+Syn	84	88	**70**	**65**
	SVM All features	91	95	59	56
	BERT	64	68	59	54

5.4. Results

The results of all our text classification experiments are presented in Table 4. Let us start with the model that we consider as our baseline, that is, using the linguistic features of the news articles belonging to Propaganda, Hoax and Satire to predict whether an article contains false or true information. Experim09Oents with an SVM using this type of approximate training data are categorized under the Rashkin mapped scenario in the table. A high training accuracy in these experiments is not surprising given both the large training data and that the validation split belonging to the same distribution of news articles. Training accuracy above 90% is indicative of a good fit to the domain data: The classifier learned very well the linguistic features appearing across all publications of trusted sources and those of the so-called fake news publishers. Now the question is whether such a model can generalize its knowledge to the detection of true from false content. Surprisingly, we see that using any of the presented feature sets, such as only the n-gram features, the model does very well in labelling fake news articles in the Buzzfeed Top dataset. However, the performance of the model drops drastically when it comes to Snopes Gold and Perez Celebrity data. We investigated the reason for this and realized that the classifier is highly biased towards labelling anything as fake news (i.e., mapped to *false*), therefore the majority of the items within the Buzzfeed Top data as well as in the other two datasets were given the false label. Given the equal number of items from both classes in the training data, one possible explanation for the classifier's bias could be that the items of the false class were a better representative of the language data that we see in the test sets; in other words, these items could have covered a larger number of topics, more varied vocabulary and writing styles. This distributional characteristic can be due to the more diverse sources of online news scraped for the fake items than for real items (mainstream trusted news) in Rashkin's data.

Overall, the low F-scores obtained on the Snopes Gold and Perez Celebrity data provides some evidence that reputation-based data collection may not be the best strategy when the target task is to detect false from true content. While the classifier seems to be good at detecting fake news (as a general label for propaganda, hoax and satire), it cannot tell when an article with a similar style is written based on facts. Therefore, the features of deception that we would like to capture for misinformation detection at the level of individual articles have not been learned in this scenario.

Next, let us examine the models that were trained on our large mixed dataset. A relatively smaller training fit is obtained in the experiments using this dataset as training material, which is expected given the smaller size of it compared to Rashkin's data. Evaluation of the models on the test datasets shows a similar pattern: Performance on Buzzfeed Top is high, but when it comes to distinguishing true news articles (in Snopes Gold and Perez Celebrity collections), the classifier shows a negative bias. This observation has a similar explanation to the one we just provided for the mapped data scenario. Most true items in the Large mix dataset come from the Buzzfeed USE dataset, which is focused on a narrow topic. Therefore, more variance will be captured for the false class with more various items in it coming from the Snopes Silver collection and the classifier would later assign the false label to the majority of test items. Recall from the previous section that this was the main reason for us to sample equal number of items from the two datasets to build a balanced training dataset.

Finally, we review the results of our experiments using the best quality training data, that is, a small but balanced set of news articles with reliable labels. In this data scenario, we get a better fit on training data and a better generalization on the test sets for all models except for BERT. Using the n-gram, semantic and syntactic features would give us the best cross-domain generalization reflected in the 70% and 65% F1-score on the Snopes Gold and the Perez Celebrity dataset. Adding the readability and surface textual features provides better results on the Buzzfeed Top items, which may have overlap in topic, headline and even entire body content with some of the training items (this can also be viewed as a type of over-fitting to the training data, which decreases the generalization power of the model).

A comparison between the classification results obtained from the BERT model and the SVM models shows the superiority of the feature-based approach with the currently available training data. Deep learning text classification techniques in general need a large amount of training data and that is what we are still lacking for the task of misinformation detection. An intriguing possibility is the application of hybrid methods, as proposed by Rohera et al. (2022), who found that a Naive Bayes algorithm obtained the highest recall in their experiments, whereas a deep learning model (LSTM) had the highest accuracy. A combination of those two methods, depending on which parameter we want to maximize, shows promise.

In summary, our results speak in favour of a hybrid approach based on both linguistic feature analysis and deep neural network models, and always taking into consideration the size and composition of the data. And, above all, the results suggest that more reliably labelled data, in the form of full news articles, is needed for this particular problem.

6. Conclusion

We have investigated the problem of misinformation in news text from a linguistic perspective, using Natural Language Processing and text classification techniques. The contributions can be summarized as the following:

- We built a dataset of false and true news articles by scraping the Snopes fact-checking pages, tracking the links to the original publisher of the news headlines and collected the body text. We also used crowdsourcing to verify the alignment between each news article and the headline labelled for veracity by the fact-checker, to make sure the data is of good quality. The Snopes Silver collection contains 1,844 texts; it has been introduced in our previous work and a small sample of it, i.e, the Snopes Gold, was used in our previous experiments (Asr and Taboada, 2019). The complete collection with crowdsourced stance data will become available upon the publication of the current manuscript.

- We analyzed the above dataset and the Buzzfeed USE dataset (from our previous work) for linguistic features indicative of false content and provided significant tests on what types of features were most discriminatory between false and true news articles.

- We conducted experiments on automatic misinformation detection using a variety of text classification techniques. By doing so, we established a new baseline for this NLP task and clarified the type of data and features that can offer the best accuracy both in within-domain and cross-domain predictions.

Our experiments show that the veracity and linguistic characteristics of a text are correlated, but high-quality training data is required to develop an accurate and scalable misinformation detection system. In particular, data should be well-distributed across topics and sources, balanced across different levels of factuality, and reliably labelled based on individual articles rather than the reputation of publishing sources, because dubious websites may publish or republish factual news articles, making the data noisy.

In terms of the machine learning techniques, we found that the classic feature-based SVM model was superior across all data scenarios. Especially in a small but balanced training data scenario, the models showed a more robust behavior, i.e., they generalized better on the test news articles with unseen headlines and claims. Based on our analysis and feature ablation study, the best features were n-grams, semantic category features and part of speech tags. Surface textual features and readability features were less effective in classification. We also found, using the same algorithm and linguistic features, that big data labelled based on reputation of the sources or big data with unbalanced topic distribution would not enhance the final system accuracy, in particular in cross-domain evaluation.

Finally, we tried a deep learning model with a pre-training phase, which helped deal with the small size of training data to some extent. However, the results show that the data we currently have is not enough to benefit from the potentials of a deep learning model. We believe that better accuracy can be obtained with an improved dataset, not only in terms of quantity but also in terms of quality. Quality improvements can be achieved with a sufficiently large collection of news articles from a variety of topics distributed in a balanced manner across the false and true target classes. Otherwise, the classifier can easily become biased towards assigning false or true labels to any unseen test item (as it was shown through our classification experiments in the larger data scenario). A larger question is whether BERT and similar deep learning architectures are truly the best tools for all classification problems. Church et al. (2021) argue that some NLP problems may be better addressed with "older rule-based systems".

Our new dataset and the properties we found for a quality dataset based on repeated experiments contribute to opening up the bottleneck in the NLP approach to misinformation detection, but more data and more contributions in the public domain are necessary. We urge researchers and, more importantly, internet and social media platforms, to share and make available such datasets for research.

Appendix A. Discriminative features in Buzzfeed USE data

Table A.5: Discriminative features in Buzzfeed data

Feature Category	Feature	Ratio false/true	P-value
Surface	num_punc/num_char	0.824111622	1.07E-07
Semantic lexicon	comparative_forms.txt	0.847129838	0.000312741
Semantic lexicon	negative_mpqa.txt	1.170890719	0.039175394
Semantic lexicon	negative-HuLui.txt	1.191929819	0.003812285
Semantic lexicon	assertives_hooper1975.txt	1.204042781	0.025607042
Semantic lexicon	factives_hooper1975.txt	1.210492088	0.075483432
Semantic lexicon	modal_adverbs.txt	1.250324586	0.08518351
Semantic lexicon	neutral_mpqa.txt	1.262506856	0.008025674
Semantic lexicon	manner_adverbs.txt	1.285541793	0.070899584
Semantic lexicon	implicatives_karttunen1971.txt	1.402833634	0.003614918
Semantic LIWC	Apostro	0.285518503	0.029453081
Semantic LIWC	i	0.597079036	0.019747465
Semantic LIWC	male	0.774133673	0.042933602
Semantic LIWC	posemo	0.84567739	0.086936016
Semantic LIWC	work	0.846851239	0.082703344
Semantic LIWC	time	0.849324655	0.008926775
Semantic LIWC	Sixltr	0.909433348	0.052975401
Semantic LIWC	relativ	0.938080123	0.059859233
Semantic LIWC	Dic	1.038763776	0.017908327
Semantic LIWC	function	1.058609143	0.000492969
Semantic LIWC	conj	1.08800601	0.026463843
Semantic LIWC	pronoun	1.0995215	0.045389199
Semantic LIWC	verb	1.12184365	0.043788628
Semantic LIWC	auxverb	1.1262107	0.091416299
Semantic LIWC	focuspresent	1.141192519	0.021681326
Semantic LIWC	adverb	1.152874365	0.009971697
Semantic LIWC	ipron	1.166344782	0.001385981
Semantic LIWC	cogproc	1.168456841	0.000735018
Semantic LIWC	insight	1.219253872	0.033014437
Semantic LIWC	quant	1.254843046	0.000564555
Semantic LIWC	we	1.300643709	0.085971532
Semantic LIWC	tentat	1.326261041	0.000128842
Semantic LIWC	certain	1.369503643	0.000657455
Semantic LIWC	bio	1.523506626	0.006427043
Semantic LIWC	they	1.669499824	4.83E-07
Semantic LIWC	health	1.763599914	0.007421396
Semantic LIWC	anx	1.936134338	0.01550405
Semantic LIWC	sexual	3.069336627	0.027490318
Syntactic	NOUN	0.934513783	0.000113056
Syntactic	DET	1.052851775	0.033674456
Syntactic	VERB	1.061155645	0.001962608
Syntactic	PRT	1.073478103	0.098291531
Syntactic	ADV	1.185689806	2.21E-05
Readability	Linsear_write_formula	0.758426991	0.000619185
Readability	Automated_readability_index	0.86367538	0.000692822
Readability	Flesch_Kincaid_grade	0.864976275	0.000797506
Readability	Gunning_fog	0.891185015	0.002323244
Readability	Smog_index	0.928396971	0.002329779
Readability	Flesch_reading_ease	1.097455733	0.004082952

Appendix B. Discriminative features in Snopes Silver data

Table B.6: Discriminative features in Snopes Silver data

Feature Category	Feature	Ratio false/true	P-value
Surface	num_char	0.832909412	0.007837336
Surface	num_punc/num_char	0.926844562	0.001331224
Surface	num_sentence	0.772084384	0.00088547
Semantic lexicon	comparative_forms.txt	0.947737051	0.020227509
Semantic lexicon	negative-HuLui.txt	1.133089752	0.00267177
Semantic lexicon	negative_mpqa.txt	1.192098625	0.000864263
Semantic lexicon	modal_adverbs.txt	1.298092007	0.005319897
Semantic lexicon	manner_adverbs.txt	1.35698137	0.001742446
Semantic LIWC	allPunc	0.648061274	0.002444304
Semantic LIWC	apostro	0.648061274	0.002444304
Semantic LIWC	money	0.69778316	0.001087094
Semantic LIWC	work	0.876750931	0.00999914
Semantic LIWC	sixltr	0.940804067	0.033111439
Semantic LIWC	article	0.953846799	0.013036009
Semantic LIWC	cogproc	1.06101936	0.057419697
Semantic LIWC	pronoun	1.070951735	0.029411617
Semantic LIWC	time	1.076314329	0.029136756
Semantic LIWC	compare	1.077072577	0.039553518
Semantic LIWC	auxverb	1.077592026	0.093201528
Semantic LIWC	adverb	1.110034072	0.00311221
Semantic LIWC	ppron	1.123737701	0.014012724
Semantic LIWC	cause	1.12805548	0.038490645
Semantic LIWC	certain	1.141792908	0.032021603
Semantic LIWC	they	1.151274939	0.066116507
Semantic LIWC	male	1.199129633	0.039660435
Semantic LIWC	bio	1.224761165	0.041183623
Semantic LIWC	body	1.328246174	0.041617951
Semantic LIWC	anx	1.383614276	0.097296795
Semantic LIWC	death	1.601757888	0.022198678
Semantic LIWC	sexual	2.753109267	0.060287064
Syntactic	NUM	0.905802493	0.06678729
Syntactic	DET	0.962340276	0.009099241
Syntactic	NOUN	0.970826348	0.021800144
Syntactic	VERB	1.034897463	0.008362629
Syntactic	PRT	1.040020625	0.097773857
Syntactic	ADV	1.093310376	0.000661848
Syntactic	PRON	1.094581886	0.011254044

Appendix C. Fine-tuned BERT model optimized hyperparameters

Table C.7: Fine-tuned BERT model hyperparameters

Hyperparameters	Tested range	Best range
Sequence Length	256 - 512	360 - 512
Number of epochs	3 - 15	4 - 12
Batch size	4 - 16	8 - 12
Dropout rate	0 - 0.5	0.25 - 0.35
Learning rate (Adam optimizer)	1E-6 - 1E-4	1E-5 - 5E-5
Warm-up steps	0 - 500	0 - 500

References

Afroz, S., Brennan, M., Greenstadt, R., 2012. Detecting hoaxes, frauds, and deception in writing style online. In: Proceedings of IEEE Symposium on Security and Privacy. San Francisco, pp. 461–475.

Allcott, H., Gentzkow, M., 2017. Social media and fake news in the 2016 election. Journal of Economic Perspectives 31, 211–236.

Asr, F. T., Taboada, M., 2018. The data challenge in misinformation detection: source reputation vs. content veracity. In: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER),Conference on Empirical Methods in Natural Language Processing. Brussels, pp. 10–15.

Asr, F. T., Taboada, M., 2019. Big data and quality data for fake news and misinformation detection. Big Data & Society, January–June 2019: 1–14.

Biyani, P., Tsioutsiouliklis, K., Blackmer, J., 2016. “8 amazing secrets for getting more clicks”: Detecting clickbaits in news streams using article informality. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence. pp. 94–100.

Church, K. W., Chen, Z., Ma, Y., 2021. Emerging trends: A gentle introduction to fine-tuning. Natural Language Engineering 27 (6), 763–778.

Conneau, A., Schwenk, H., Barrault, L., LeCun, Y., 2017. Very deep convolutional networks for text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Valencia, pp. 1107–1116.

Connolly, K., Chrisafis, A., McPherson, P., Kirchgaessner, S., Haas, B., Phillips, D., Hunt, E., Safi, M., December 2 2016. Fake news: An insidious trend that’s fast becoming a global problem. The GuardianHttps://www.theguardian.com/media/2016/dec/02/fake-news-facebook-us-election-around-the-world.

Conroy, N. J., Rubin, V. L., Chen, Y., 2015. Automatic deception detection: Methods for finding fake news. In: Proceedings of the Conference of the Association for Information Science and Technology. Vol. 52. St. Louis, pp. 1–4.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. Minneapolis, pp. 4171–4186.

Dos Santos, C., Gatti, M., 2014. Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics. Dublin, pp. 69–78.

Eisenstein, J., 2019. Introduction to Natural Language Processing. MIT Press, Cambridge, MA.

Fichtner, U., December 20 2018. Der Spiegel reveals internal fraud. Der Spiegel.
URL http://www.spiegel.de/international/zeitgeist/claas-relotius-reporter-forgery-scandal-a-1244755.html

Fogg, B. J., Marshall, J., Laraki, O., Osipovich, A., Varma, C., Fang, N., Paul, J., Rangnekar, A., Shon, J., Swani, P., Treinen, M., 2001. What makes web sites credible? A report on a large quantitative study. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 61–68.

Guo, Z., Schlichtkrull, M., Vlachos, A., 2022. A survey on automated fact-checking. Transactions of the Association for Computational Linguistics 10, 178–206.

Habgood-Coote, J., 2019. Stop talking about fake news! Inquiry 62 (9-10), 1033–1065.

Horne, B. D., Adali, S., 2017. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. arXiv preprint arXiv:1703.09398.

Horne, B. D., Khedr, S., Adali, S., 2018. Sampling the news producers: A large news and feature data set for the study of the complex media landscape. In: Proceedings of the Twelfth International AAAI Conference on Web and Social Media. Palo Alto, pp. 518–527.

Hovy, D., 2020. Text Analysis in Python for Social Scientists. Cambridge University Press, Cambridge.

Kucharski, A., 2016. Post-truth: Study epidemiology of fake news. Nature 540 (7634), 525–525.

Lai, S., Xu, L., Liu, K., Zhao, J., 2015. Recurrent convolutional neural networks for text classification. In: Proceedings of AAAI Conference on Artificial Intelligence. pp. 2267–2273.

Lazer, D., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S. A., Sunstein, C. R., Thorson, E. A., Watts, D. J., Zittrain, J. L., 2018. The science of fake news. Science 359 (6380), 1094–1096.

Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning. Beijing, pp. II–1188–II–1196.

Lugea, J., 2021. Linguistic approaches to fake news detection. In: Deepak, P., Chakraborty, T., Long, C., Kumar, S. G. (Eds.), Data Science for Fake News: Surveys and Perspectives. Springer, Cham, pp. 287–302.

Mantzarlis, A., December 18 2017. Not fake news, just plain wrong: Top media corrections of 2017. Poynter News.
URL https://www.poynter.org/news/not-fake-news-just-plain-wrong-top-media-corrections-2017

Medvedeva, M., Kroon, M., Plank, B., 2017. When sparse traditional models outperform dense neural networks: The curious case of discriminating between similar languages. In: Proceedings of the 4th Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial). Valencia, pp. 156–163.

Mukherjee, A., Venkataraman, V., Liu, B., Glance, N., 2013. Fake review detection: Classification and analysis of real and pseudo reviews. Technical Report UIC-CS-2013–03, University of Illinois at Chicago, Tech. Rep.

Ng, A., 2011. Why is Deep Learning taking off? Tech. rep., Coursera.
URL https://www.coursera.org/lecture/neural-networks-deep-learning/why-is-deep-learning-taking-off-praGm

Nørregaard, J., Horne, B. D., Adalı, S., 2019. NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In: Proceedings of AAAI Conference on Web and Social Media. Munich, pp. 630–638.

Ouyang, X., Zhou, P., Li, C. H., Liu, L., 2015. Sentiment analysis using convolutional neural network. In: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. IEEE, pp. 2359–2364.

Pennebaker, J. W., Boyd, R. L., Jordan, K., Blackburn, K., 2015. The development and psychometric properties of LIWC 2015. Technical report, University of Texas at Austin.

Pennington, J., Socher, R., Manning, C. D., 2014. Glove: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Doha, pp. 1532–1543.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R., 2017. Automatic detection of fake news. arXiv preprint arXiv:1708.07104.

Pérez-Rosas, V., Mihalcea, R., 2015. Experiments in open domain deception detection. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Lisbon, pp. 1120–1125.

Perrott, K., 2016. A fake news on social media influenced US election voters, experts say. ABC 26.
URL http://www.abc.net.au/news/2016-11-14/fake-news-would-have-influenced-us-election-experts-say/8024660

Post, M., Bergsma, S., 2013. Explicit and implicit syntactic features for text classification. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Vol. 2. Sofia, pp. 866–872.

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., Stein, B., 2018. A stylometric inquiry into hyperpartisan and fake news. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, pp. 231–240.

Przybyla, P., 2020. Capturing the style of fake news. In: Proceedings of AAAI Conference on Artificial Intelligence. Vol. 34. New York, pp. 490–497.

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., Choi, Y., 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, pp. 2921–2927.

Recasens, M., Danescu-Niculescu-Mizil, C., Jurafsky, D., 2013. Linguistic models for analyzing and detecting biased language. In: Proceedings of the Conference of the Association for Computational Linguistics. Sofia, pp. 1650–1659.

Riggins, S. H., 1997. The rhetoric of othering. In: Riggins, S. H. (Ed.), The language and politics of exclusion. Sage, Thousand Oaks, CA, pp. 1–30.

Rohera, D., Shethna, H., Patel, K., Thakker, U., Tanwar, S., Gupta, R., Hong, W. C., Sharma, R., 2022. A taxonomy of fake news classification techniques: Survey and implementation aspects. IEEE Access 10, 30367–30394.

Rubin, V. L., Chen, Y., Conroy, N. J., 2015. Deception detection for news: Three types of fakes. In: Proceedings of the Conference of the Association for Information Science and Technology. Vol. 52. St. Louis, pp. 1–4.

Rubin, V. L., Conroy, N. J., Chen, Y., Cornwell, S., 2016. Fake news or truth? Using satirical cues to detect potentially misleading news. In: Proceedings of NAACL-HLT. San Diego, pp. 7–17.

Ruchansky, N., Seo, S., Liu, Y., 2017. CSI: A hybrid deep model for fake news detection. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. Singapore, pp. 797–806.

Ruths, D., 2019. The misinformation machine. Science 363 (6425), 348–348.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., Liu, H., 2020. FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big Data 8 (3), 171–188.

Silverman, C., Strapagiel, L., Shaban, H., Hall, E., Singer-Vine, J., October 20 2016. Hyperpartisan Facebook pages are publishing false and misleading information at an alarming rate. BuzzFeed Newshttps://www.buzzfeed.com/craigsilverman/partisan-fb-pages-analysis.

Tandoc Jr, E. C., Lim, Z. W., Ling, R., 2018. Defining “fake news”: A typology of scholarly definitions. Digital Journalism 6 (2), 137–153.

Thorne, J., Vlachos, A., Christodoulopoulos, C., Mittal, A., 2018. FEVER: A large-scale dataset for Fact Extraction and VERification. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, LA, pp. 809–819.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., Polosukhin, I., 2017. Attention is all you need. In: Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Vol. 30. Curran Associates, Inc., pp. 5998–6008.
URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Volkova, S., Shaffer, K., Jang, J. Y., Hodas, N., 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on Twitter. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vol. 2. Vancouver, pp. 647–653.

Vosoughi, S., Roy, D., Aral, S., 2018. The spread of true and false news online. Science 359 (6380), 1146–1151.

Wang, W. Y., 2017. ‘Liar, liar pants on fire’: A new benchmark dataset for fake news detection. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vol. 2. Vancouver, pp. 422–426.

Wardle, C., Derakhshan, H., 2017. Information disorder: Toward an interdisciplinary framework for research and policy making. Report, Council of Europe.

Wilson, T., Wiebe, J., Hoffmann, P., 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. Association for Computational Linguistics, pp. 347–354.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J., 2019. Huggingface's transformers: State-of-the-art natural language processing. ArXiv abs/1910.03771.

Yang, F., Mukherjee, A., Dragut, E., 2017. Satirical news detection and analysis using attention mechanism and linguistic features. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhaguen, pp. 1979–1989.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E., 2016. Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Vancouver, pp. 1480–1489.

Zhang, A. X., Ranganathan, A., Metz, S. E., Appling, S., Sehat, C. M., Gilmore, N., Adams, N. B., Vincent, E., Lee, J., Robbins, M., 2018. A structured response to misinformation: Defining and annotating credibility indicators in news articles. In: Proceedings of the The Web Conference 2018. International World Wide Web Conferences Steering Committee, Lyon, France, pp. 603–612.

Zhang, X., Zhao, J., LeCun, Y., 2015. Character-level convolutional networks for text classification. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. Montréal, pp. 649–657.